Abstract

The amount of software in many systems increases exponentially. This increase impacts the reliability of these systems. In the source code of software many hidden faults are present. These hidden faults can transform into errors during the system life cycle, due to changes in the system itself or in the context of the system.

We will discuss the current trends and potential directions for future solutions of an increasing reliability problem.
Introduction
+ ESI
+ Speaker

Exploration
+ trends
+ consequences for reliability
+ potential solutions

Research Projects
+ Trader
+ Tangram
+ Ideals
+ Boderc

conclusion
Mission
To advance industrial innovation and academic excellence in embedded systems engineering (ESE).

Vision
ESI and its partners create and apply world-class ESE methods.

7 Founders:
Industry (Philips, ASML, Océ)
Universities (Twente, Delft, Eindhoven)
Knowledge Institutes (TNO)

cardio X-ray system
television
GSM
waferstepper
printer
Embedded Systems Engineering

performance

reliability

evolvability

in relation to all other system qualities: security, power, cost, size, et cetera
Industry as Laboratory

- **Source of inspiration**
- **Application playground**
- **Industry**

**Challenging problems**
- **Apply new engineering methods**
- **Research**
- **Hypothesis**

**Evaluate results**
- **Improve**
Who is Gerrit Muller?

CT OSS NM X-ray

MR

EasyVision

Gaudí

CT OSS NM X-ray

Philips Medical

ASML

1980 1990 2000

industrial experience
time pressure pragmatics sales
cost constraints products
lots of people

research
reflection evidence
exposure
education

Buskerud
Exploration of Reliability

Introduction
  + ESI
  + Speaker

Exploration
  + trends
  + consequences for reliability
  + potential solutions

Research Projects
  + Trader
  + Tangram
  + Ideals
  + Boderc

conclusion
Trends in Embedded Systems

How to survive in innovative domains?

- Fast moving market
- Fast moving technology
- Complex value chains
- Increased integration
Increased Team Size

- **Historic Trend**: A linear increase in required team size from 1995 to 2010.
- **Our Challenge**: A less steep increase compared to the historic trend.

<table>
<thead>
<tr>
<th>Year</th>
<th>Required Team Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>10</td>
</tr>
<tr>
<td>2000</td>
<td>100</td>
</tr>
<tr>
<td>2005</td>
<td>1000</td>
</tr>
<tr>
<td>2010</td>
<td>1000</td>
</tr>
</tbody>
</table>
Number of Faults Proportional With Code Size

Based on average 3 errors/kloc

Reliability of Software Intensive Systems
11 Gerrit Muller
Hard Reset Required:

- Cell Phone
- Television
- PC
- Beamer
- Car
- Coffee Machine
- DVD player
- Airbus

Pilot announces a flight delay, due to computer problems. A complete reset is required. The flight entertainment system also show a reset: a complete Linux boot. This reboot hangs: server xxx not found
How to Make SW Intensive Systems Reliable

- within cost&time
- improve reliability
- without performance penalty
- with increasing requirements
- increasing #suppliers #sites, et cetera

- reduce functionality
- more/longer testing
- more procedures
- improve way-of-working
- reduce code
- more standardization
- improve recovery
- improve detection
- improve robustness

- more formality
- more automation
- shared understanding
- early integration
- more feedback
- tolerant interfaces
- robust patterns

Reliability of Software Intensive Systems
13  Gerrit Muller

version: 0.2
September 6, 2020
RSISolutions
Introduction
  + ESI
  + Speaker

Exploration
  + trends
  + consequences for reliability
  + potential solutions

Research Projects
  + Trader
  + Tangram
  + Ideals
  + Boderc

Conclusion
Example Trader Project

Overview

Trader  Television Related Architecture and Design to Enhance Reliability
Objective: Develop method & tools to ensure reliable consumer electronics products
Research agenda: System Reliability
Domain: Digital TV
CIP: Philips Semiconductors
Partners: Philips CE, Tass, and Research
DTI, IMEC, TUD, TU/e, UL, UT, ESI

Industrial Relevance

Poor reliability has severe business impact
- Customer expectation of TV reliability is high
  - Little tolerance for technical problems
- 100% fault-free design is not achievable
- High volume market implies high risks if reliability problems occur
  - Low product margin leaves no buffer for service costs
  - Service center costs multiplied by number of complaints
  - Market share reduction likely, i.e. customers buy another brand
- (On) Time to market is critical
  - Missing fixed shipping gates costs millions of dollars

User Perceived Reliability

- Objective
  - Determine the user-perceived severity of a product failure mode
- Methodology
  - Create a model considering relevant factors
    - User-perceived loss-of-functionality
    - User-perceived reproducibility
    - Failure-frequency
    - Work-around difficulty
    - User-group characteristics
    - Failure characteristics
    - ...
  - Validate model
  - Evaluate and suggest system failure recovery strategies
    - The recovery strategy may not annoy the user even more!

Trader Domain Trends

TV complexity increase follows the PC world
from “Display movies over antenna” to “Display anything over everything”

Broadcast
- ATSC, DVB, ISDB, Analog
- Terrestrial, Satellite, Cable

Connectivity
- Cameras, JPEG, flash cards, HDD, MP3, Web browser, Ethernet, USB, etc...

User-Perceived Reliability

Aspects (all depends on user)

<table>
<thead>
<tr>
<th>Aspects</th>
<th>1</th>
<th>2</th>
<th>3 (JPEG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Importance</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Frequency</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Reproducibility</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Solvability</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Loss of Function / Time / Behavior</td>
<td>4</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>576</td>
<td>160</td>
<td>0</td>
</tr>
</tbody>
</table>

Reliability of Software Intensive Systems
15  Gerrit Muller

version: 0.2
September 6, 2020
RSIStrader
Increasing Code Size in Televisions

Moore's law

From: COPA tutorial, Rob van Ommering

1965

1979

1990

1 kB

64 kB

2 MB

Reliability of Software Intensive Systems

version: 0.2
September 6, 2020
LWAmooreLawRvO
Research: System Awareness to Improve Reliability

user input → system → system output

monitor

system failure model

awareness

correction

customer expectations

system output
Exposé product weaknesses at design time

- **Source code analysis**
  - Identify hotspots in code
  - Consider impact to user-visible behavior: *prioritize warnings*

- **Software architecture reliability analysis**
  - Techniques to identify failure-prone components
  - Evaluation of architectural alternatives and trade-offs
Quality Degradation Caused by Shit Propagation

- needed code
- code not relevant for new function
- repair code
- new needed code
- copy
- paste
- modify
- bad code
- new bad code

Reliability of Software Intensive Systems
19  Gerrit Muller
Example of Shit Propagation

Class Old:
    capacity = startCapacity
    values = int(capacity)
    size = 0

def insert(val):
    values[size]=val
    size+=1
    if size>capacity:
        capacity*=2
        relocate(values, capacity)

def insertBlock(v,len):
    for i=1 to len:
        insert(v[i])

Class New:
    capacity = 1
    values = int(capacity)
    size = 0

def insert(val):
    values[size]=val
    size+=1
    capacity+=1
    relocate(values, capacity)

Class DoubleNew:
    capacity = 1
    values = int(capacity)
    size = 0

def insert(val):
    values[size]=val
    size+=1
    capacity+=1
    relocate(values, capacity)

def insertBlock(v,len):
    for i=1 to len:
        insert(v[i])
Integration & Test

Early Integration

- Incremental
- Model Based
- Model <-> Realization
- Infrastructure
- NDDS, Matlab, Chi

Results:
- Early Model Based Integration and Testing in AGILE
- Problem found in Sn CoMo before review
Moore’s Law

Reliability of Software Intensive Systems

version: 0.2
September 6, 2020
ASMLmooreLaw
Challenge: Exponential Increase

- Performance
- Feedback and adjustments
- Imaging
- Overlay
- Productivity
- HW and SW components
- Innovation
- Reliability
- Robustness
- Complexity
4 Views on a Waferstepper

**subsystems**

- Laser light source
- Illuminator beam shaping
- Reticle stage positioning
- Reticle handler input/output
- Lens projection
- C&T contamination, temperature
- Wafer stage positioning
- Wafer handler input/output

**control hierarchy**

- System control coordination
  - Laser
  - Illuminator
  - Lens
  - Measurement
  - C&T
  - Reticle stage
  - Reticle handler
  - Wafer stage
  - Wafer handler

- Ethernet

**physics/optics**

- Kinematic
  - Slit
  - Reticle
  - Wafer
  - Dynamic exposure through slit

- NA abberations
  - Transmission
  - Aerial image
  - Lens
  - Uniformity
  - Pulse-freq, bw, wavelength, ...

Reliability of Software Intensive Systems

version: 0.2
September 6, 2020

EBMSystemDiagrams

24  Gerrit Muller
Balancing functionality, quality and time/cost
(over 20% reduction of integration & test time)

Dynamic simulation of integration & test approach:
1. Create model (modules, interfaces, faults, tests)
2. Execute model at any time
3. Balancing based on time, cost and remaining risk.
Example Ideals Project

code size \[\text{reduce by}\] refactoring \[\rightarrow\] cross cutting concerns

Results: Parameter Check

- Automatic replacement of current parameter checking and tracing idiom with specific aspect oriented idiom.

- Code size reduction of 80% for parameter checking idiom (7% reduction of total module)

- Improved locality
Example Boderc Project

31x5E  
2050  
2090
Boderc Goal

Boderc goal =

A specific methodology
to predict

multidisciplinary
system performance

within industrial constraints and restricted design space

based on modeling
and analyze, discuss, document, and communicate
throughput, quality

people, process, project duration, and cost

power computing response time

Reliability of Software Intensive Systems
version: 0.2
Gerrit Muller
September 6, 2020
BS06bodercGoal
Shared Understanding by Modeling

- System
- Multi-disciplinary
- Mono-disciplinary

Number of details:

- Back-of-the-envelop
- Formula based
- Executable

Reliability of Software Intensive Systems

Gerrit Muller
version: 0.2
September 6, 2020
Many Models Needed to Understand System

6. Kinematic modeling

7. Thermo modeling
8. Control architecture
9. Virtual printer models
10. Stepper motors

small, simple, goal-driven models
shorter cycle time, less cycles
shorter product creation lead time
Coverage of Reliability by ESI Projects

- within cost & time
- improve reliability
- without performance penalty
- with increasing requirements
- increasing #suppliers #sites, et cetera

- more/longer testing
- more procedures
- reduce functionality
- improve way-of-working
- reduce code
- improve recovery
- improve detection
- improve robustness
- more formality
- more automation
- early integration
- shared understanding
- more feedback
- tolerant interfaces
- robust patterns

Boderc
Tangram
Ideals
Trader
Trader
Trader
RSiScoverage
version: 0.2
September 6, 2020
Towards a Conclusion, Some more Trends

Introduction
  + ESI
  + Speaker

Exploration
  + trends
  + consequences for reliability
  + potential solutions

Research Projects
  + Trader
  + Tangram
  + Ideals
  + Boderc

Conclusion
Applications depend on chain of systems

- Users
- Network Providers
- Service Providers
- Content Providers
- Home Server
- Infotainment appliance
  - Watch video
  - Browse photo's
  - Calendar
  - And much more...
Interoperability: systems get connected at all levels

Reliability of Software Intensive Systems
34 Gerrit Muller

version: 0.2
September 6, 2020
Multi dimensional interoperability

<table>
<thead>
<tr>
<th>Integrating multiple applications</th>
<th>in multiple languages</th>
<th>delivered by multiple vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>clinical analysis</td>
<td>clinical support</td>
<td>Philips</td>
</tr>
<tr>
<td>administrative</td>
<td>USA, UK, China, India</td>
<td>GE</td>
</tr>
<tr>
<td>financial</td>
<td>Japan, Korea</td>
<td>Siemens</td>
</tr>
<tr>
<td>workflow</td>
<td>France, Germany</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Italy, Mexico</td>
<td></td>
</tr>
</tbody>
</table>

Based on multiple media, networks:
- DVD+RW
- memory stick
- memory cards
- bluetooth
- 11a/b/g
- UTMS

And multiple standards:
- Dicom
- HL7
- XML

And multiple releases:
- R5
- R6.2
- R7.1
Interoperability Trends and Research Challenges

**trends**
- market dynamics
- interoperability
- reliability

**(partial) solutions**
- standards
- design patterns

**new research challenges!**
- globalisation
- hype waves
- Moore's law
- emerging behavior
- future vs legacy
- heterogeneous vendors
- dynamics (continually changing)
- complexity
- heterogeneity
- dynamics
- #engineers involved
- economical interests
- latency due to slow process
- most fundamental solution
- in system feedback
- human-in-the-loop feedback
- semantic understanding
- containment
- gateway
- ....

---

Reliability of Software Intensive Systems

version: 0.2
September 6, 2020
RSISInteroperability
Based on average 3 errors/kloc less code
reduce impact of hidden faults on system reliability
less errors/kloc
1000
10k
1000
100
10
100 kloc
100 Mloc
10Mloc
manyyears and LOC (lines of code) per product

Conclusion